Revista Ciência Agronômica, v. 51, Special Agriculture 4.0, e20207741, 2020
Centro de Ciências Agrárias - Universidade Federal do Ceará, Fortaleza, CE
www.ccarevista.ufc.br

Artigo Científico
ISSN 1806-6690

Development of a robotic structure for acquisition and classification of images (ERACI) in sugarcane crops

Desenvolvimento de estrutura robótica para aquisição e classificação de imagens (ERACI) em cultura da cana-de-açúcar

José Ricardo Ferreira Cardoso[1]*, Carlos Eduardo Angeli Furlani[2], José Eduardo Pitelli Turco[2], Cristiano Zerbato[2], Franciele Morlin Carneiro[2] and Francisca Nivanda de Lima Estevam[2]

ABSTRACT - Digital agriculture contributes to agricultural efficiency through the use of such tools as computer vision, robotics, and precision agriculture. In this study, the objective was to develop a system capable of classifying images through the recognition of pre-established patterns. For this purpose, a geographically distributed system was created, based on the Raspberry Pi 3B+ computer, which captures images in the field and stores them in a database, where they are available to receive a pre-classification by a supervisor. Subsequently, classifiers are generated, evaluated, and sent to the remote device to conduct a classification in real time. For an evaluation of the system, 23 classes were defined and grouped into 3 superclasses, 36,979 images were captured, and 1,579 pre-classifications were conducted, which allowed the classification tests to be carried out by means of a cross-validation by randomly dividing into the equivalent number of classes. These tests revealed that the accuracy delivered by each classifier is different and directly proportional to the quantity and balance of the samples, with a variation of 11% to 79%, with 26 and 2,200 samples considered, respectively. The response time of the system was evaluated during 1,585 periods and was maintained within approximately 0.20 s, and under controlled speed of the vehicle, can be used for the dispersion of inputs in real time.

Key words: Digital Agriculture. Machine Learning. Open source. Raspberry Pi. Computer Vision.

RESUMO - A agricultura digital contribui com a eficiência agrícola por meio da utilização de ferramentas como a visão computacional, robótica e agricultura de precisão. Com este trabalho o objetivo foi desenvolver um sistema capaz classificar imagens por meio do reconhecimento de padrões pré estabelecidos. Para este fim foi criado um sistema distribuído geograficamente, baseado no computador Raspberry Pi 3B+, que captura imagens no campo e armazena em um banco de dados, onde estão disponibilizadas para receber uma pré-classificação por parte de um supervisor. Depois disso, classificadores são gerados, avaliados e enviados para o dispositivo remoto realizar a classificação em tempo real. Para a avaliação do sistema foram definidas 23 classes agrupadas em 3 superclasses, capturadas 36.979 imagens e, realizadas 1.579 pré-classificações, que permitiram a realização de testes de classificação por meio de validação cruzada com divisão equivalente a quantidade de classes e de forma embaralhada. Estes testes mostraram que a acurácia entregue por cada classificador é diferente e, diretamente proporcional a quantidade e balanceamento das amostras, com variação da acurácia de 11% a 79%, com 26 e 2.200 amostras consideradas, respectivamente. O tempo de resposta do sistema foi avaliado em 1.585 períodos e se mantiveram em aproximadamente 0,20 segundos, podendo, sob velocidade controlada do veículo, ser utilizada para dispersão de insumos em tempo real.

Palavras-chave: Agricultura digital. Machine Learning. Open source. Raspberry Pi. Visão computacional.

DOI: 10.5935/1806-6690.20200102
Editores do artigo: Professor Daniel Albiero - daniel.albiero@gmail.com e Professor Alek Sandro Dutra - alekdutra@ufc.br
*Author for Correspondence
Received for publication 15/01/2020; approved on 24/11/2020

[1] Área de Informática, Instituto Federal de Educação, Ciência e Tecnologia de São Paulo/IFSP, Avenida C-Um, 250, Residencial Ide Daher, Barretos-SP, Brasil, 14.781-501, jrfercar@ifsp.edu.br, (ORCID ID 0000-0001-6476-6898)

[2] Departamento de Engenharia e Ciências Exatas, Faculdade de Ciências Agrárias e Veterinárias/FCAV, Universidade Estadual Paulista/UNESP, Jaboticabal-SP, Brasil, eduardo.furlani@unesp.br (ORCID ID 0000-0002-1508-5372), jose.turco@unesp.br (ORCID ID 0000-0001-5227-3738), cristiano.zerbato@unesp.br (ORCID ID 0000-0002-4534-5454),franmorlin1@gmail.com.br (ORCID ID 0000-0003-0117-7468), nivanda_lima@hotmail.com (ORCID ID 0000-0002-0667-348X)

INTRODUCTION

Despite the difficulties generated by the cost of production, mainly concentrated in the harvest, sugarcane is an important crop in Brazil owing to demand for products generated through its processing (CONAB, 2020).

Waste from the inputs is considered when producing the crop. For this reason, to reduce such waste, and consequently the production costs, while maintaining the good development of the culture, the use of equipment with information systems capable of carrying out the necessary actions in the places previously, has spread. determined. (BERNARDES; BELARDO, 2015).

The use of technologies in the context of precision agriculture allows the mapping of soil fertility and the development of culture with localized management intervention (RESENDE; COELHO, 2017). Equipment that obtains information on plant health, such as a canopy reflectance sensor and normalized difference vegetation index sensors, which present responses in real time, has been used more frequently in the field and is one of the most promising options for supplying a technological bottleneck (CASTRO, 2016). In addition to these technologies, devices interconnected by the network (IoT) show a clear trend, together with the development of remote sensing equipment (GIMENEZ; MOLIN, 2018).

In addition, the recognition of the plant of interest is important to be able to distinguish whether the application of a certain nutrient or pesticide should occur (KHMAG *et al.*, 2017). To this end, the leaves of the plants have proven to be viable sources of information and are widely used in the identification of different species. This task is usually conducted by specialists, which, given the need for high productivity in the field, is infeasible.

Within the context of agriculture technologies, several scientific studies have used digital image processing and artificial intelligence techniques for the capture, segmentation, extraction of characteristics, and classification of images (GONZALEZ; WOODS, 2010; LUGER, 2013). According to Kang and Oh (2018), the means by which images are obtained deserves special attention, and one choice may imply a high performance of the system while sacrificing user convenience, whereas another choice can bring about great convenience, but with a performance reduction of the user system.

The concern with the rapid development of new technologies, i.e., eliminating technology-based tasks already on the market, has led to a diffusion and use of software, framework, and opensource hardware (OSROOSH *et al.*, 2018; MISHRA *et al.*, 2019; SAHU *et al.*, 2019).The overall objective of the present research was the implementation of a robotic structure capable of detecting the presence of sugarcane and weeds, as well as the absence of plants. The following specific objectives were therefore stipulated: a) acquire and store images in a database, b) allow the generation of computer knowledge through computer vision algorithms and manage such knowledge, c) check the relationship, number of samples versus accuracy delivered by machine learning algorithms, d) apply pattern recognition in real time, and e) analyze the response time of the system, from image acquisition to classification.

MATERIAL AND METODHS

The Robotic Structure for Image Acquisition and Classification (ERACI) was designed and developed in an integrated effort between the Agricultural Machinery and Mechanization Laboratory (LAMMA) and the Instrumentation, Acquisition, and Processing Laboratory (LIAP), both from FCAV/UNESP and campus Barretos of the Federal Institute of São Paulo between August 2018 and December 2019. Image capture and device tests were conducted in the municipality of Barretos-SP, located around the geographic coordinates of latitude 20 ° 33 ‘33″ S, longitude 48 ° 34 8″ W and altitude 544, 0 m. According to the Köppen classification, the climate of the region is classified as Aw (ALVARES *et al.*, 2013). The average annual rainfall is 1,309 mm, with maximum and minimum temperatures of 24.5 °C and 19.4 °C, respectively (CLIMATE-DATA, 2020). The soil presents well-diversified characteristics passing through the clayey purple eutrophic latosol on the bank of the Rio Pardo and sandy red dystrophic Latosol to Argisol in the western part of the municipality (INPE, 2020).

The diversity of images captured by ERACI was aimed at adapting the system and conducting an evaluation under different types of algorithms in the computer vision and machine learning area, and images of rural and urban environments that can be grouped into superclasses such as urban and rural, and more specifically, sugarcane crops were included (Table 1). Altogether, the DRR traveled, captured images, and was tested within an area of approximately 17.2 ha.

The system was designed to work under a modular architecture, with its parts geographically and independently separated, while guaranteeing security, scalability, and transparency. These characteristics allow the exchange of information between the devices that compose it, in addition to connecting several specialists through mobile devices (DMs) with a generator and knowledge manager device (DGGC) and conducting a pre-classification of images stored in the database (Figure 1).

In the installation diagram (Figure 1), the remote robotic device (DRR) node corresponds to the structure taken to the field to conducted the image capture and tests. It is connected to a node called a *router*, which is a TP-Link® router, model TL-MR 3420, capable of allowing connection to the Internet through another router through an RJ-45 wide area network (WAN) port or through a 3G/4G modem connected to a universal serial bus (USB) port. When the DRR is connected to the router, it is possible to make use of a mobile device connected to the same router for control and monitoring in real time through the "eraci" application.

The DGGC (Figure 1) has software capable of storing and managing the data of images, classes, user, and classifications performed by the user exercising the role of the teacher of the system, within the context of supervised learning of machine learning. In addition, the DGGC has the ability to maintain hypertext transfer protocol (HTTP) services and generate files based on machine learning algorithms, capable of applying pattern recognition to images. The device represented by this node is also connected to a TP-Link router; however, the WR-840N model only allows connection to the Internet through its RJ-45 WAN gateway.

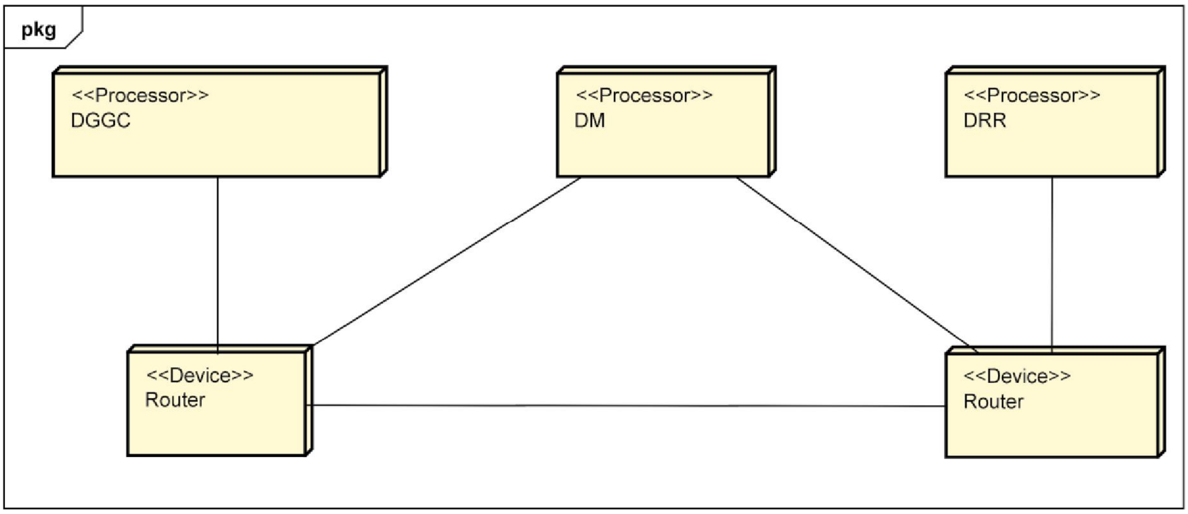
When both extreme devices, DRR and DGGC, are in operation and connected to their respective routers they are, in turn, connected to each other through a wired network or the Internet, allowing them to exchange information. In this case, the DRR can send the captured images to the DGGC, which processes them, extracts the characteristics, and stores the images and characteristics in the database. The DGGC sends the classifier files to the DRR, allowing it to recognize the patterns in the images that are being "viewed."

The DM node corresponds to a mobile device that has an installed ERACI application. Using this application, it is possible to manage the data stored in the database, classify the images, control the viewing angle, and monitor in real time what the DRR is "seeing" and how the images are being classified. The exchange of information between the DRR and the DGGC occurs through transmission control protocol/Internet protocol (TCP/IP) sockets for text data with control and monitoring information, and the user datagram protocol/Internet protocol (UDP-based IP) for image data sent by the DRR.

The way in which the system is organized allows several instances of DRR and DM, and the interaction between them occurs through the computer network using HTTP, TCP/IP, and UDP/IP. Within a client/server architecture, each DRR and DM behaves as a client connected to the DGGC, which is capable of receiving numerous connections from both clients. Thus, it is possible that, in a production environment, there may be several DRRs loading images to a single server capable of generating increasingly better ML models. To generate this "knowledge", it is also possible that there are several users classifying images for use with ML algorithms.

Each diagram node (Figure 1) has a series of software components, with the most diverse built-in functionalities, which are divided into DRR subsystems, DGGC subsystems, and DM application subsystems. All of these subsystems were implemented in Java and Python and run on Android and Raspbian environments.

Figure 1 - System installation diagram showing the devices involved in each of the nodes and how they interact with each other



To obtain patterns in the images, the characteristics that were considered, namely, the aspect (mode, mean, and standard deviation of the image converted into gray scale), dimension (area and perimeter), and inertial data (24 moments of the image and 7 invariant moments, as described by Hu (2012)).

The generation of the image to be classified followed the procedures below:

Obtain an image within the RGB color space;

Obtain the mode of the image;

Obtain the standard deviation of the image converted into grayscale;

Convert the image into the HSV color space;

Segment the image by color within the HSV color space centered on the mode, with the minimum and maximum thresholds equivalent to the standard deviation:

a. Minimum threshold = Mode − Standard deviation;

Maximum threshold = Mode + Standard deviation.

Detect the edges using Canny edge detection;

Obtain the center of mass of the image;

Obtain the dimensional average of the image;

Draw a circle with a diameter proportional to 10% of the segmented image area;

Insert a cross at the point corresponding to the center of gravity of the image with a size proportional to 10% of the diameter of the circle.

For the generation of classifiers, the following procedures are applied:

Loading of pre-classified samples;

Training of the classifiers, based on the K-nearest neighbors (KNN), support vector machine (SVM), logistic regression (RL), naive Bayes (NB), decision tree (DT), random forest (RF), and multilayer perceptron (MLP) algorithms;

Evaluation of the accuracy of each classifier by means of the cross-validation method, with the division equivalent to the number of classes existing in the iteration and by the "accuracy_score" function of the "scikitlearn" framework;

Determine the accuracy of the classifier by averaging the accuracy results;

Determine the standard deviation of the accuracy;

Storage of the classifier in the file.

The constantly changing sunlight and overlapping of the research object are two factors that significantly influence the image recognition, adding complexities such as color, brightness, and shape (GONZALEZ; WOODS, 2010). Because the way the images are captured by the camera directly affects the performance of the system (KANG; OH, 2018), it was considered that the characteristics extracted from the image will be used for the previous classification by a supervisor, and therefore preference was given to the reduction of the system response time and simplification of the capture method, to the detriment of the application of more robust pre-processing and feature extraction techniques, such as those applied by Hao *et al.* (2019), which achieved an accuracy above 94%.

The generation of a new classifier was defined to be conducted for every 50 new samples pre-classified by the supervisor. This is due to the hypothesis that the methodology used for the classification of the images is capable of allowing an accuracy proportional to the quantity, and to the balance of the samples used to generate the classifiers.

Table 1 - Classes and superclasses used for the classification of a given image

Superclass 1	Class 1	Superclass 2	Class 2	Superclass 3	Class 3
Rural Area	Weed	Sugarcane crops	Sugarcane	Urban Area	Tree
	Tree		Weed		Vehicle
	Ground		Ground		Edification
	Heaven		Heaven		Sky
	Person		Tree		Person
	Vehicle		Edification		Weed
	Edification		Vehicle		Ground
	Agriculture		Person		

The classification of an image follows the procedures below:

Extraction of image characteristics;

Choosing which classifier presented the best accuracy for classifying the superclass;

Estimation of the superclass that best represents the image;

Choice of which classifier presents the best accuracy to classify the class;

Estimation of which class best represents the marked area of the image;

Estimation of the accuracy delivered by the system using simple averages (Equation 1) below.

$$acc = \frac{accsupc + accsubc + pp}{3} \quad (1)$$

Where *acc* is the estimated accuracy, *accsupc* is the accuracy of the superclass classifier, *accsubc* is the accuracy of the class classifier, and *pp* is the result of the “predict_proba” function abstracted from the “scikitlearn” framework. Thus, each of the classifiers received values by referring to the parameters presented in Table 2 and as output values relative to those presented in Table 1. The classifiers SVM, RL, and NB used the standard configuration specified by the framework. The classifiers KNN, DT, RF, and MLP were structured as listed below, with the missing parameters maintaining their values standardized by the framework:

K-Nearest Neighbors (KNN): The neighborhood used is determined at run time with its value being an odd number, from 1 to the number of possible classes for a sample.

Decision Tree (DT): This was parameterized to use the entropy method and establish the nodes of the tree.

Random Forest (RF): The number of trees used is defined at runtime, and its value can vary between 1 and 40. It has also been parameterized to use the entropy function and establish the nodes of each tree.

Multilayer Perceptron (MLP): The best configuration for the MLP to be used by the algorithm was achieved through trial and error, and it was shown that, to reach the best classification result, it was convenient to use 1000 iterations, with a learning rate of 0.000010 and 4 hidden layers with 400 neurons each.

To verify whether the algorithms are learning as the samples are inserted, a process was created at the DGGC that adds 50 samples to each iteration in a cyclic generation function and extracts those values related to accuracy under this determined quantity of data. This quantity was stipulated at random, and aims to allow the generation of data that describe the evolution, or not, of the accuracy of the algorithms.

In the accuracy verification process, cross-validation was used with the number of divisions equivalent to the number of classes present in each iteration, and scrambled the samples in each of them (LISKI *et al.*, 2020). The choice of this method for checking the accuracy is due to the number of samples in each iteration being different and unbalanced.

The DRR was implemented on a Raspberry Pi (RPI) computer, and the connection between it and each of the modules is through the general-purpose input/output (GPIO) pins.

To avoid exposing the hardware to any damage with regard to, mainly, the accidental disconnection of a cable, a plastic structure was wrapped around it while maintaining the accessibility to the buttons, pan-tilt structure, camera, infrared reflector, power bank, HDMI terminal, and terminals for powering the RPI and charging the sealed 12V battery. In addition, holes were created to allow access to the memory card, RJ45, and USB terminals. These holes were closed with plastic plates screwed into the fairing of the robot (Figure 2).

To capture the images, the DRR was attached to a 2008 Volkswagen Fox by pressing it into the side window of the left side door (driver’s side). In an interesting Thus, the parts used by the infrared reflector and camera are on the outside of the vehicle, and the remaining parts, such as the buttons and the light-emitting diode (LED) indicator, remain on the inside. The distance between the ground, at a flat location, and the camera of the device remained at approximately 1.35 m, and a variation of up to 0.15 m may have occurred because of existing irregularities in the terrain when capturing the image.

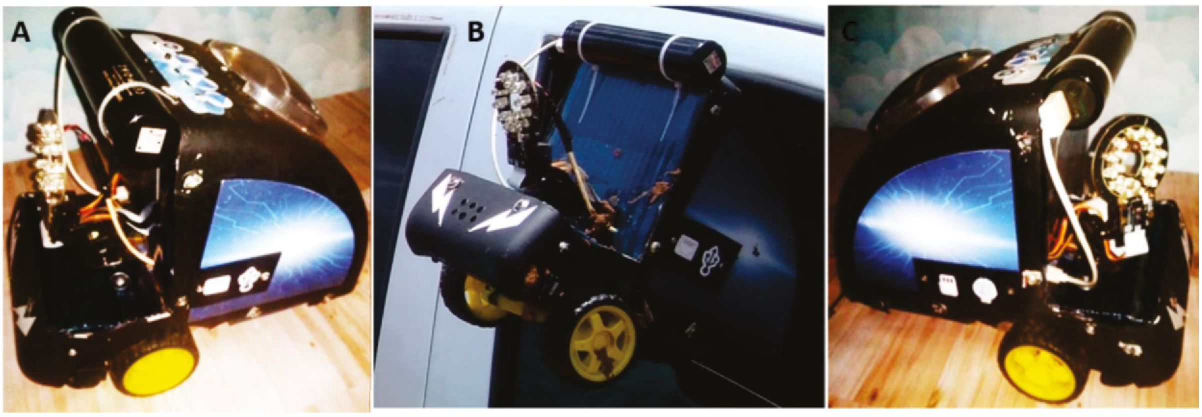
The DGGC was implemented on a Raspberry Pi model 3B + computer and was configured to store the data necessary for the generation of knowledge and its management. A HAT module with a 3.7 V lithium battery and 3800 mAh, capable of managing the energy supply and guaranteeing an operating autonomy of up to 3 h, was used without power from the power outlet. This strategy was applied to mitigate the corruption of data stored on the RPI memory card owing to fluctuations and power outages. To help reduce the temperature in the RPI, a heatsink with a cooler tower type was added.

The DRR is made up of software that runs as a service of the Raspbian operating system, which is capable of allowing, at runtime, settings to be made

Table 2 - Fields and respective types of data used for training and testing the algorithms

Number	Field	Type	Variation
1	Red Mode	Integer	0 – 255
2	Green Mode	Integer	0 – 255
3	Blue Mode	Integer	0 – 255
4	Red Mean	Float	0 – 255
5	Green Mean	Float	0 – 255
6	Blue Mean	Float	0 – 255
7	Gray mean	Float	0 – 255
8	Red standard deviation	Float	0 – 255
9	Green standard deviation	Float	0 – 255
10	Blue standard deviation	Float	0 – 255
11	Gray standard deviation	Float	0 – 255
12	hu1	Float	∞
13	hu2	Float	∞
14	hu3	Float	∞
15	hu4	Float	∞
16	hu5	Float	∞
17	hu6	Float	∞
18	hu7	Float	∞

Figure 2 - Robot with plastic structure for protection



regarding the DGGC addressing, communication port numbers, security passwords, form of image capture, and task performance, as shown in Table 3 (CARDOSO, 2020). The configuration data required for the device to function are stored in a text file on the device itself in encrypted form using the MD5 technique (LANDGE; SATOPAY, 2018).

To allow an analysis of the time spent by the DRR to conduct all processes related to computer vision, using the images captured by the camera to the recognition of patterns with the generation of new images with information regarding the classification, a method was added to this subsystem that receives the time elapsed between these two milestones and stores it in a text file.

Similar to the DRR, the DGGC is composed of software initialized as a service of the Raspbian operating system. These are responsible for the execution of tasks, such as data storage, provisioning of HTTP services, the extraction of image characteristics, and the generation of classifiers using machine learning techniques.

Table 3 - DRR subsystems

Subsystems	Function
Robot startup	This loads the other subsystems needed for the device to function.
Control	This shuts down and restarts the system, captures images and video clips, and moves the robot and positions the camera.
Monitoring	This monitors the operating time, low-voltage alert, date and time (hour and minute) when the last data storage took place, and the global positioning, and if necessary, activates the control subsystem to prevent a device shutdown.
Archiving	This verifies at 60-s intervals new files stored in the image and video directories, and when connected to the Internet, transfers the files to the database located at the DGGC.
Communication	Whenever available, this makes a validated password connection with the DGGC and DM and allows image data, classifiers, real-time supervision, and device control to be exchanged.
Vision	This captures images using the camera, extracts their characteristics and submits them to the classifier algorithm. From the recognition of the image superclass and the associated item with the center of mass of the segmented image, a new image is generated and is available to be sent to the supervisory system installed in the DRR, with information regarding the superclass, class, and estimated accuracy, according to (1).

Postgree is used as the database management system and is structured within the relational paradigm, enabling security restrictions. The data stored there include images, characteristics, users, classes, and classifications. To allow a manipulation of the data stored in the database by the systems involved, all developed within the paradigm of object-oriented programming (OOP), the SQLAlchemy framework is used to map the records stored in the objects and vice versa (CARDOSO, 2020).

For information protection reasons, all data transfer between devices connected to the DGGC takes place through HTTP services, such as GET, POST, PUT, and DELETE, with authentication requirements and in accordance with the Representational State Transfer architecture (CARDOSO, 2020).

To ensure that the database can be restored and thus guarantee the security and integrity of the data, a shell script was created allowing the data stored to be periodically backed up. Likewise, a shell script was created to purge these backups to mitigate the probability of overloading the capacity of the data storage unit, intended for this purpose.

An application called “eraci” was developed running on the Android operating system. Its functionalities are related to data management, image pre-classification, and DRR supervision. In this way, this application mitigates the risks related to the protection of image data and pre-classifications carried out by an unauthorized person. This application receives a marked image of the web services subsystem, where the user can apply its exclusion or classification. In addition to these features, it is composed of a local supervisory subsystem capable of real-time viewing regarding how the image is being classified by the DRR, in addition to enabling the positioning of the camera, as needed.

The pre-classification of the images takes place by the user connected to the system through the “eraci” application installed in the DM, along a call to be made to the web service subsystem in the DGGC. This service sends the applicant a marked image, obtained at random.

RESULTS AND DISCUSSION

The development of the system under the distributed systems architecture made it possible to establish, through the “eraci” application installed in the DM, the possible classes to which an image can belong. In this way, the classes were organized into superclasses and classes, with the purpose of allowing a given class to belong to a superclass that best represents a group of images (Table 1). These superclasses allowed the grouping of classes categorically into rural area, sugarcane crops, and urban area.

Through the DRR, it was possible, between September 4, 2019 and January 24, 2020, to capture 36,979 images, among which images of sugarcane were captured at different stages of development. Between December 20, 2019 and February 25, 2020, 2,200 images were pre-classified, and distributed at rates of 44% (urban area), 43% (sugarcane crops), and 13% (rural area). With the data generated using the cross-validation accuracy verification process, graphs were generated that showed the relation between the number of samples and the accuracy of the classification algorithms (Figures 3-9).

As can be seen in the graphs (Figures 3-9), the minimum quantity for regularity (MQR) occurs along with the lowest accuracy delivered by the respective algorithm or global minimum, and remains in constant growth, with the occurrence of fluctuations, in which the minimum iteration does not present an accuracy of less than that presented in the MQR.

Based on the hypothesis that the use of a random classification methodology would provide (100/7 ~ 14%) and (100/8 ~ 12%) chances for a correct classification to be carried out. It was considered that, to ensure that the algorithm recognizes patterns and make decisions based on the algorithms, the accuracy must be greater than 14% for the urban area and greater than 12% for the rural and sugarcane crop areas. It is reasonable to expect that these levels of accuracy will increase as the number of samples increases, along with the trend of quantitative equilibrium.

From the data resulting from the estimation of accuracy, it was found that, initially, owing to the low variability of the categories, the accuracy of the classifiers proved to be greater. However, as the number of categories increased, there was a greater difficulty in applying the classifications, resulting in a lower accuracy delivered by the algorithms. The analysis of the graphs also showed a constant increase in accuracy from a certain amount of data, with oscillations proportional to the increase in categories, and in a marked way, to the oscillations, when using the KNN and RF algorithms according to the number of neighbors and trees used, respectively. For this reason, greater importance was given to discovering the amount of pre-classified data, for which the algorithms begin to show significant accuracy.

For this purpose, information for each categorical group was gathered in Tables 4-8, i.e., information regarding the point on the graph where positive and regular growth of the accuracy occurs. It was verified that this point is equivalent to the global minimum of the function, which relates the accuracy to the quantity of the samples, and is called the MQR.

When the MQR is analyzed in terms of the classification of superclass, it should be noted that the MLP classifier reaches this point with 1,250 samples, with an accuracy close to 74% and an approximate standard deviation of 0.084 (Table 4). From this point on, variations in accuracy occur. In the 2,200 samples, the approximate accuracy is 79% with a standard deviation of 0.007.

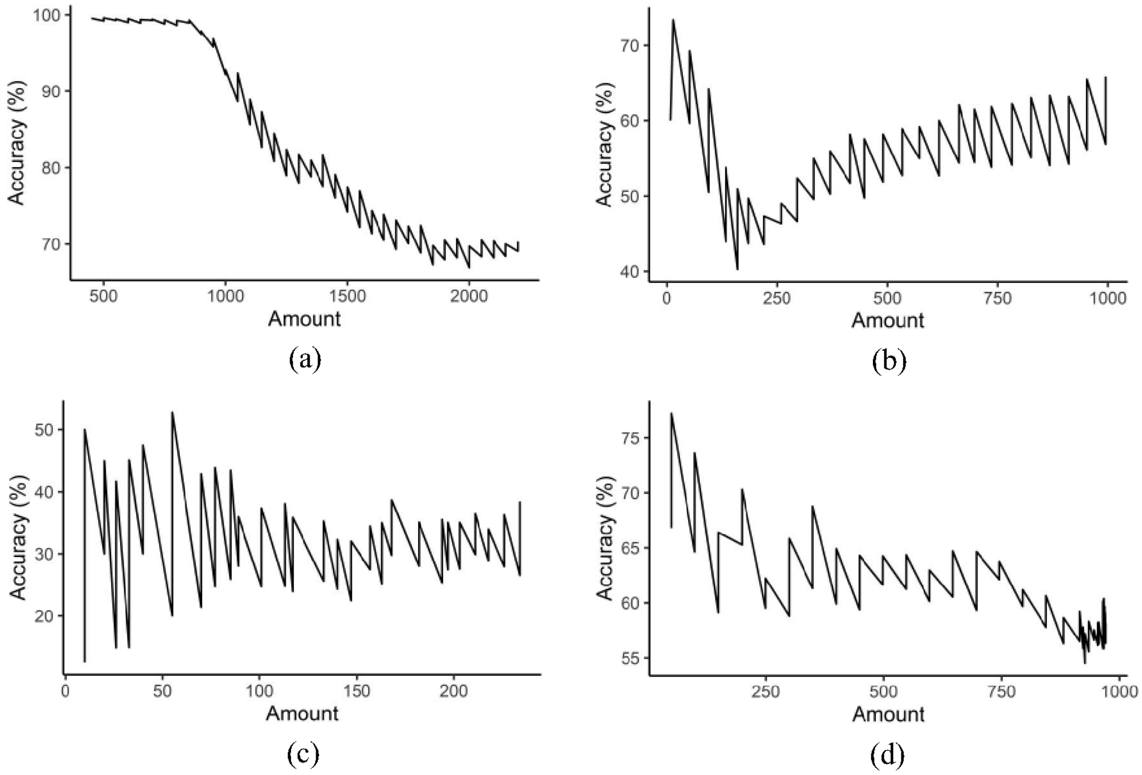
For this purpose, information on each categorical group was gathered in Tables 4-8, including information regarding the point on the graph where positive and regular growth in accuracy occurs. It was verified that this point, i.e., the MQR, is equivalent to the global minimum of the function, which relates the accuracy with the number of samples.

When the MQR is analyzed in terms of the classification of the superclass, it should be noted that the MLP classifier reaches this point with 1,250 samples, with an accuracy of close to 74% and an approximate standard deviation of 0.084 (Table 4). From this point on, variations in accuracy occur. In all 2,200 samples, the approximate accuracy is 79% with a standard deviation of 0.007.

When the MQR is observed in terms of the classification of the classes belonging to the sugarcane group, this point is reached by the RF classifier, with approximately 36% accuracy and a standard deviation of 0.063. When all samples are considered, its accuracy is close to 69% with a standard deviation of 0.029. Table 5 shows that, although the MLP is not the classifier that reaches this point first, it reaches it with 185 samples with better accuracy than that delivered by the RF. A negative point regarding the MLP for this result is that it presents a standard deviation with a higher value, thus showing greater uncertainty regarding the forecasts.

When the analysis is conducted in terms of the classification of the classes belonging to the rural area cluster, it has to start at the beginning, with an MQR equal to 10 (Table 6). At this value, the KNN, MLP, and NB classifiers start the classification on a regular basis. The accuracy presented by the NB classifier is disregarded because it is unable to conduct the classification at this

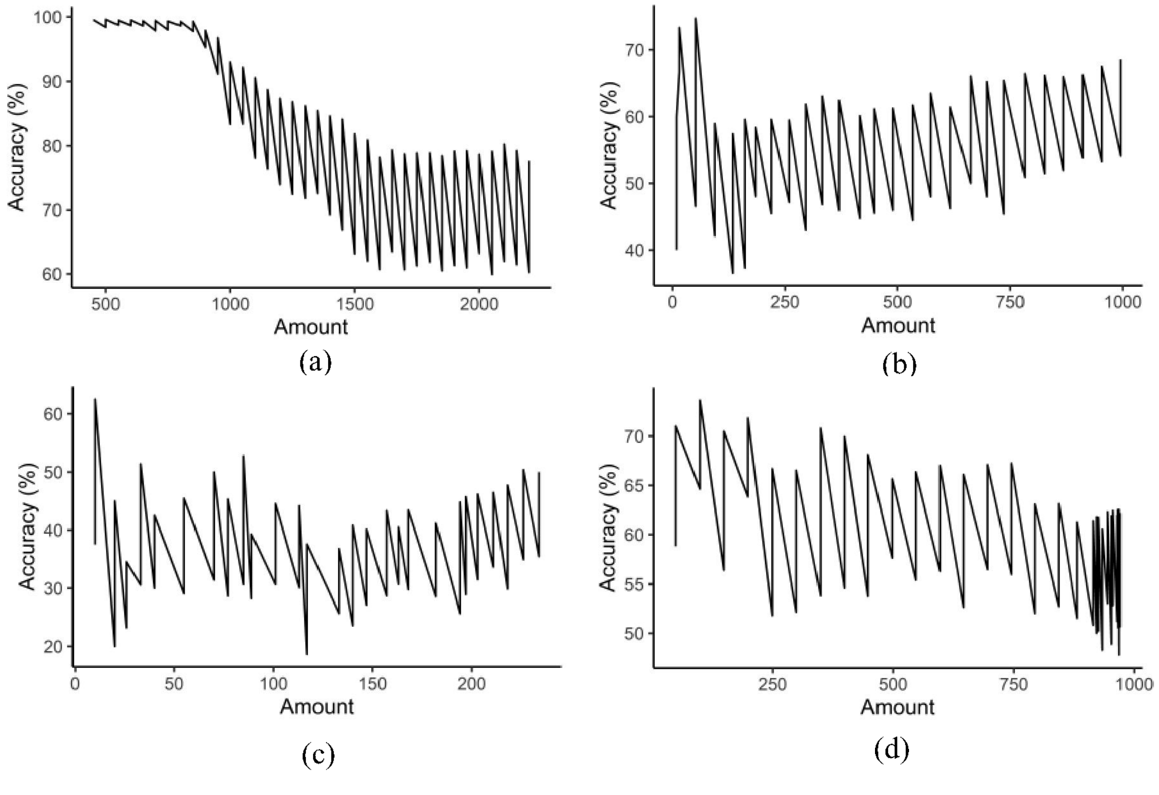
Figure 3 - Relation between quantity of samples and accuracy of the KNN algorithm. (a) superclass and (b) sugarcane crops, (c) rural area, and (d) urban area classes



(a) (b)

(c) (d)

Figure 4 - Relation between number of samples and accuracy of the RF algorithm: (a) superclass and (b) sugarcane crop, (c) rural area, and (d) urban area classes



(a) (b)

(c) (d)

Figure 5 - Relation between number of samples and accuracy of the DT algorithm: (a) superclass and (b) sugarcane crop, (c) rural area, and (d) urban area classes

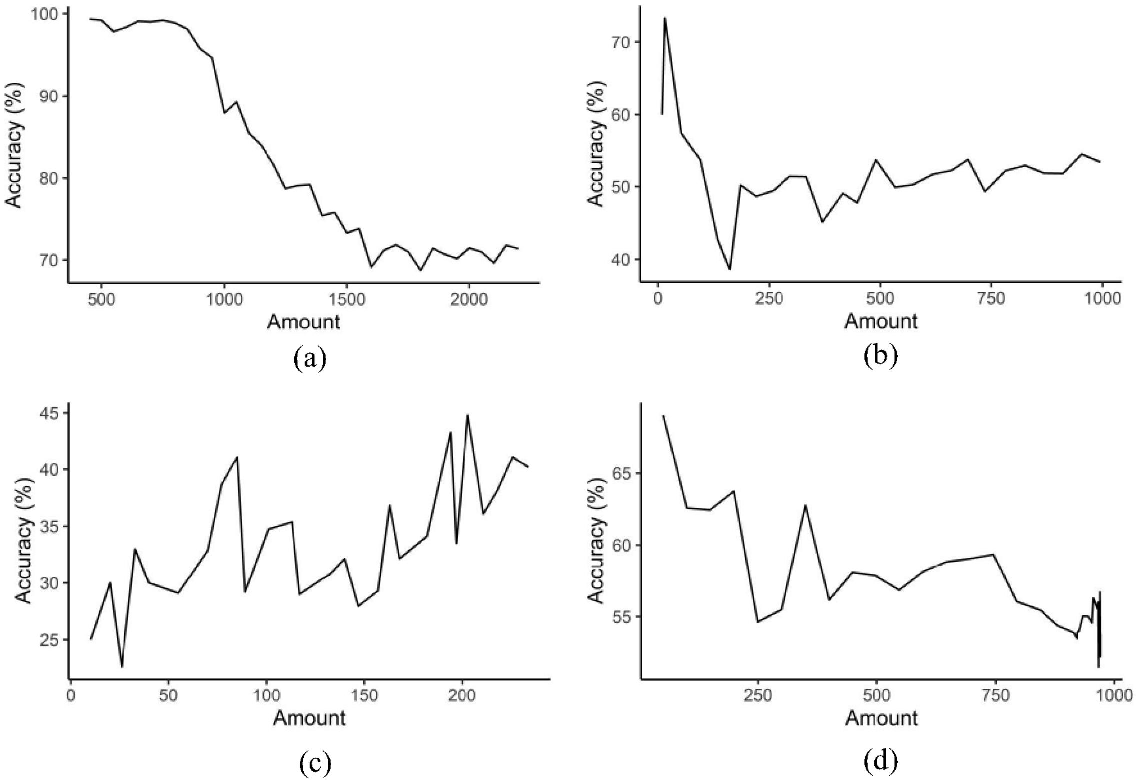
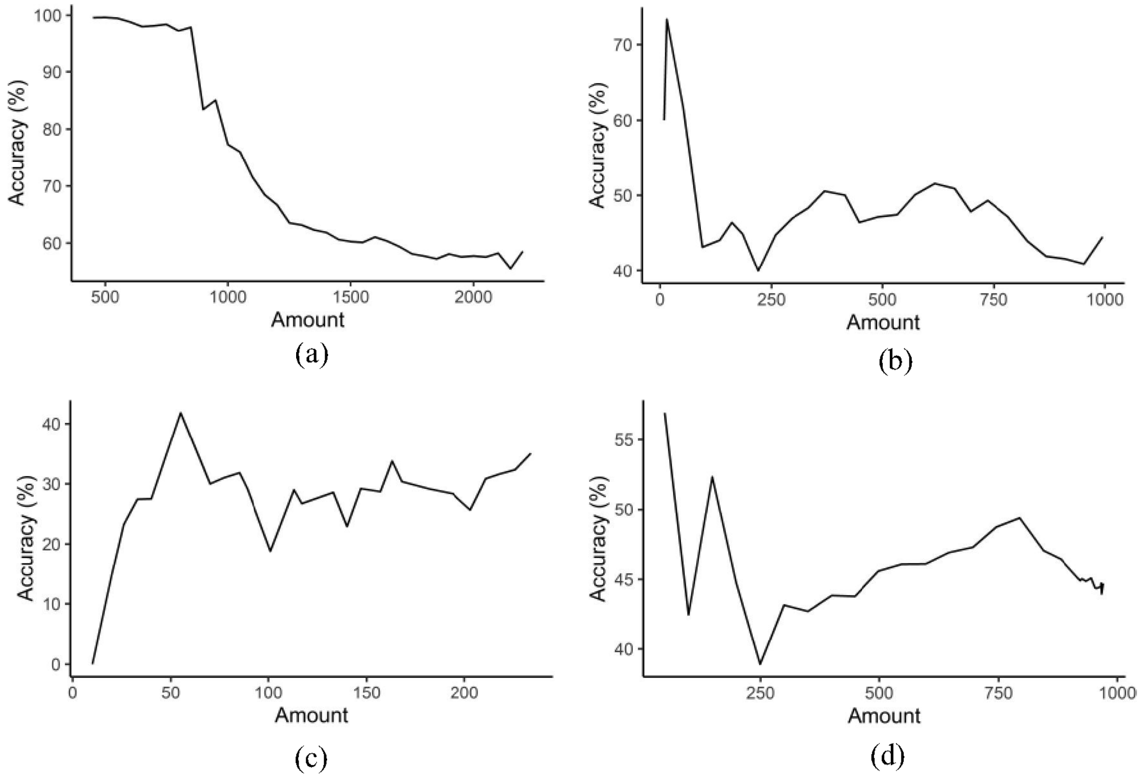


Figure 6 - Relation between number of samples and accuracy of the NB algorithm: (a) superclass and (b) sugarcane crop, (c) rural area, and (d) urban area classes



point; however, it shows, along with the other classifiers, an evolution in its accuracy (Figure 6c).

Considering the accuracy delivered by these three algorithms, KNN and MLP (Figure 3c and Figure 9c) show a similar positive evolution, with an accuracy of 12.5% and a standard deviation of approximately 0.21 (Figure 6). Although this accuracy is not the desired level because it is equivalent to the accuracy obtained when using any means of random classification, it shows a significant improvement, with fluctuations and the minimum value during each iteration reaching greater than 12.5%.

The accuracies of the NB, KNN, and MLP algorithms for all 234 samples were 35% (standard deviation of 0.091), 39% (standard deviation of 0.066), and 39% (standard deviation of 0.087), respectively.

The NB classifier was the first to achieve the RMS for classifying the urban area cluster classes with 249 samples (Table 7). The accuracy at this point was approximately 38%, and the standard deviation was approximately 0.064. When all samples were used, the accuracy was 45% and the standard deviation was 0.0379.

One of the reasons for the oscillation in accuracy throughout the process of increasing the samples for training is the imbalance in relation to the number of categories during each process (RUSTOGI; PRASAD, 2019). This fact allows the algorithms to become extremely good at recognizing patterns related to a certain category, and poor at others having fewer samples.

Through an analysis of these data, it is possible to state based on the residual mean square that the methods applied to conduct the classification of the images have a tendency to achieve a better accuracy. Thus, the accuracy delivered by the algorithms tends to increase proportionally with the increase in the number of pre-classified records.

Owing to the variation in accuracy delivered by the algorithms, it can be stated that the strategy of checking at runtime which algorithm delivers the highest accuracy for the classification is valid because, in this way, the classifier that best suits the number of pre-classified data is always used for training.

Using the data from the file generated by the proposed method and applied to a computer vision

Figure 7 - Relation between number of samples and accuracy of the RL algorithm: (a) superclass and (b) sugarcane crop class, (c) rural area, and (d) urban area classes

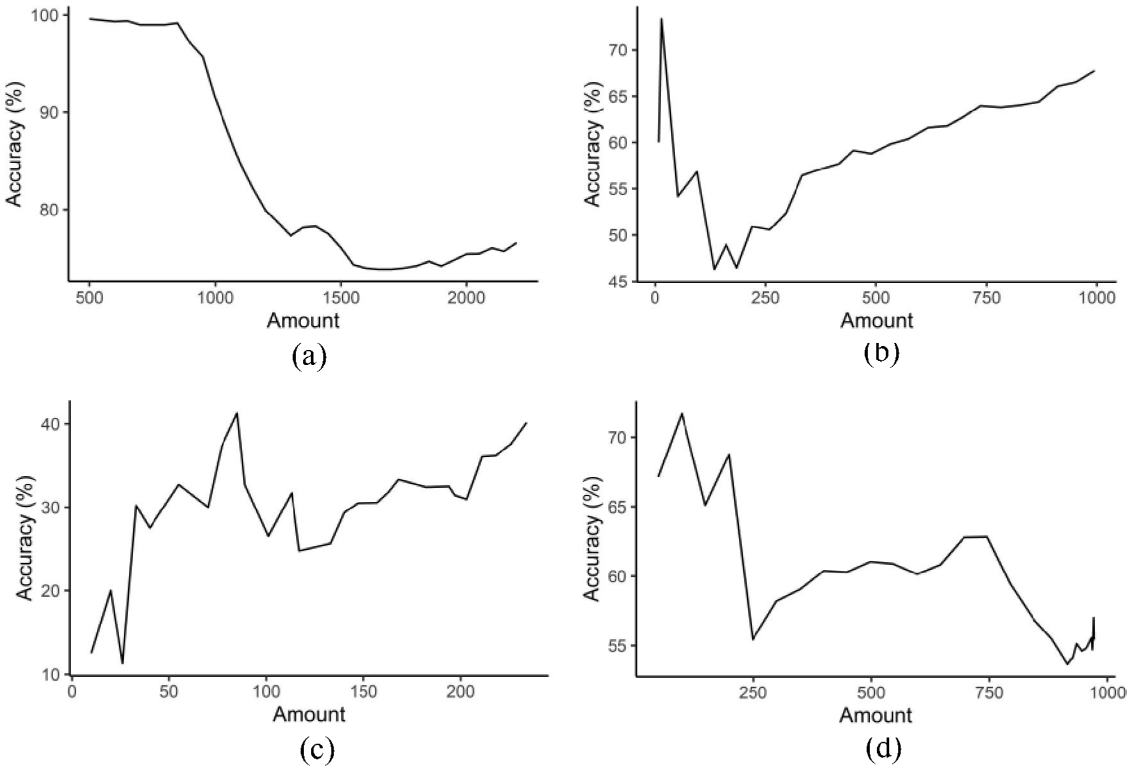
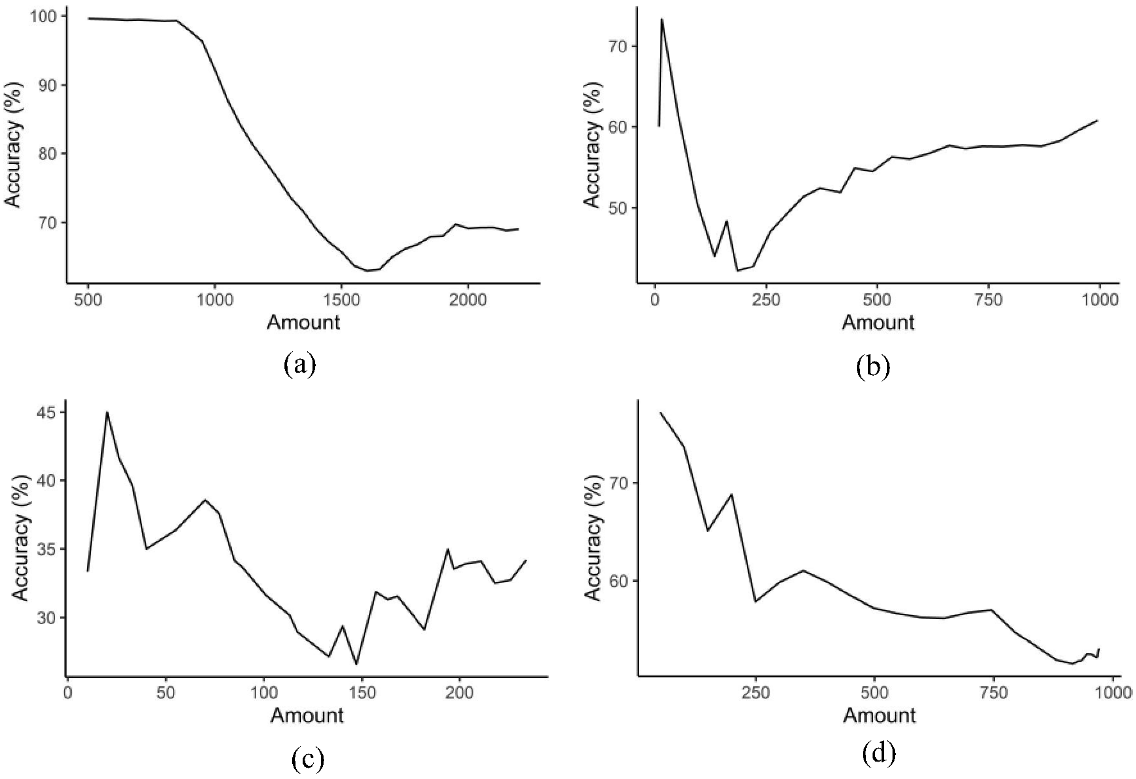


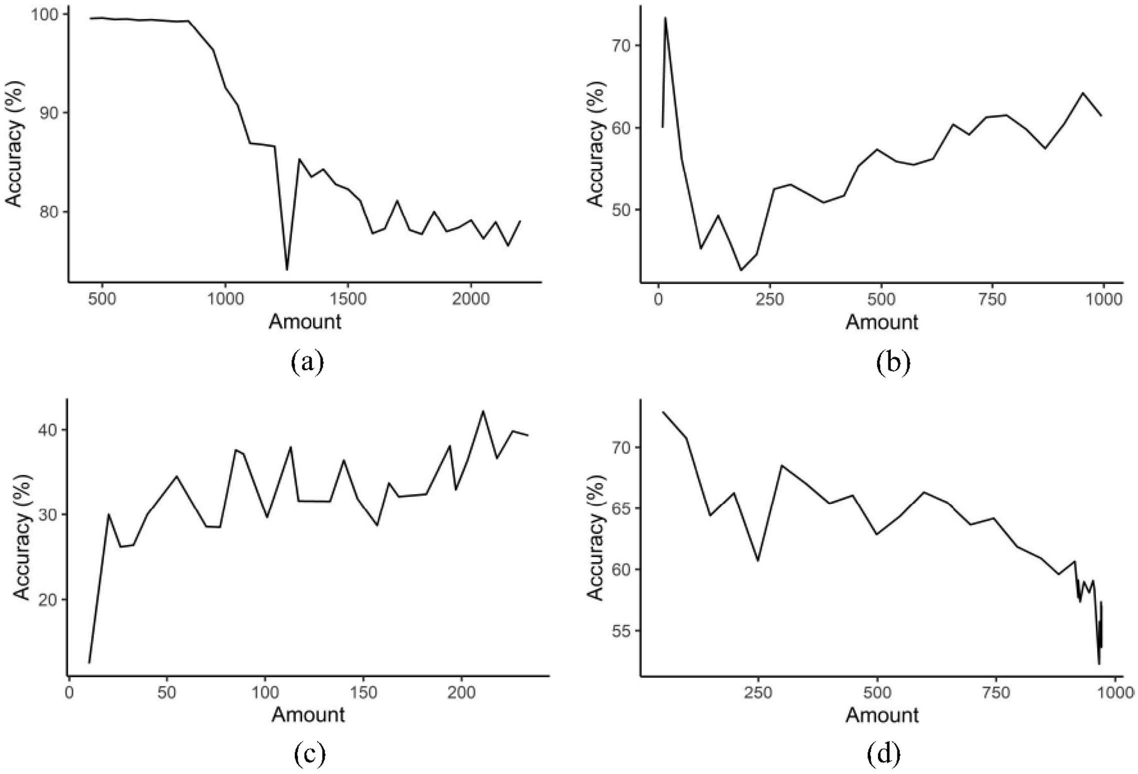
Figure 8 - Relation between the number of data and the accuracy of the SVM algorithm: (a) superclass and (b) sugarcane crop, (c) rural area, and (d) urban area classes



(a) (b)

(c) (d)

Figure 9 - Relation between number of data and accuracy of the MLP algorithm: (a) superclass and (b) sugarcane crop, (c) rural area, and (d) urban area classes



(a) (b)

(c) (d)

Table 4 - Residual mean square related to classifiers for the classification of superclass

Classifier	QMR	Accuracy	Standard deviation
DT	1800	68,72222222	0,007737993
KNN	2000	66,90078384	0,011677682
MLP	1250	74,0803511	0,084252265
NB	2150	55,53425846	0,011706561
RF	2050	59,90313346	0,011707259
RL	1650	73,87878788	0,013388316
SVM	1600	62,99735556	0,036370276

Table 5 - Residual mean square related to the classifiers for the classification of the sugarcane crop classes

Classifier	QMR	Accuracy	Standard deviation
DT	161	38,53276353	0,085595626
KNN	161	40,26590693	0,134091588
MLP	185	42,6344086	0,108922396
NB	220	39,96496496	0,0708332
RF	134	36,52832675	0,063101137
RL	134	46,31093544	0,108465841
SVM	185	42,16845878	0,067352286

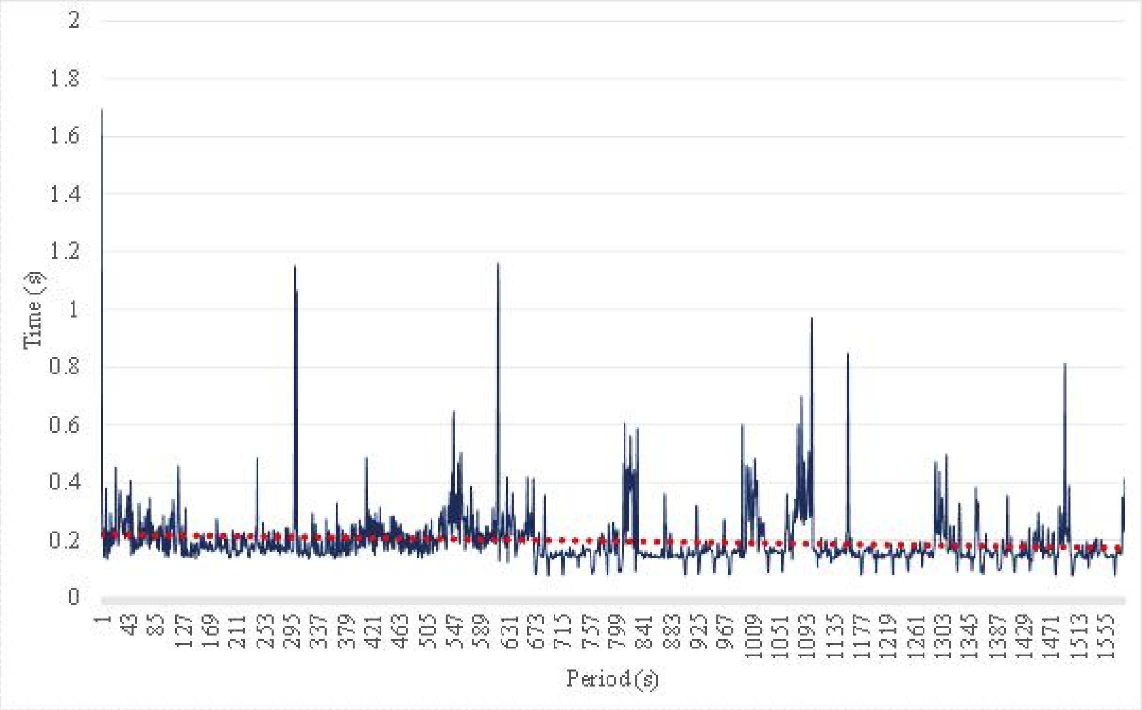
Table 6 - Residual mean square when classifying rural area classes

Classifier	QMR	Accuracy	Standard deviation
DT	26	22,61904762	0,132030197
KNN	10	12,5	0,216506351
MLP	10	12,5	0,216506351
NB	10	0	0
RF	117	18,64035088	0,087667748
RL	26	11,30952381	0,120673422
SVM	147	26,58333333	0,109196349

Table 7 - Residual mean square when classifying urban area classes

Classifier	QMR	Accuracy	Standard deviation
DT	967	51,49975718	0,039464116
KNN	926	54,52869711	0,034832129
MLP	966	52,27743271	0,058860741
NB	249	38,90824623	0,064327257
RF	969	47,79605347	0,050603489
RL	915	53,67145396	0,045285085
SVM	915	51,48922142	0,060668246

Figure 10 - Processing time required to apply all procedures for extracting the characteristics and recognizing patterns in the image captured in real time using the DRR over a period of 1.585 s



subsystem, a graphic was generated that allows an analysis of the values (Figure 10).

An analysis of the data that served as the basis for the generation of the response time graph (Figure 10) showed that the time spent for processing the data, from capturing the image to recognizing the patterns within the image, varies by approximately 0.0757 and 1.6956 s, which gives the device an average processing time of 0.196 s, with a standard deviation of 0.0992 s.

The trend line of the graph, shown by the red dashed line, shows that the response time tends to be close to 0.2 s, which ensures that the system can be used to conduct real-time classifications. However, the results suggest that it can be used in production only when the speed of the vehicle, to which the DRR is attached, is controlled by the DRR itself, which should allow the classification of a new image only after a classification has been completed.

CONCLUSION

1. ERACI was made using open source resources, with software available for free and low-cost computer-based hardware focused primarily on computer education. Its development demonstrated that the Raspberry Pi computer can be expanded for the development of more robust projects, serving not only as a prototyping plate, but also as a processing unit for robotic systems aimed at productive sectors, and is thus an alternative low-cost computing resource;
2. The system made it possible to carry out the acquisition and storage of the images captured in a database in an automated and intelligent manner, because it is able to detect when a possible connection occurs between the DRR capturing the image and the DGGC storing the image in the database;
3. Another aspect desired for the system is the ability to generate knowledge through computer vision algorithms, in addition to being able to manage this knowledge. These objectives were achieved through the DGGC module, which conducts the extraction of the characteristics and the generation of knowledge in a cyclical and automatic manner. In addition, it provides web services for the management and sharing of generated knowledge;
4. Pattern recognition in images was achieved through the DRR, which is able to capture images, send them to the DGGC, obtain the classifiers through access to a service published by the DGGC, and conduct

the pattern recognition in real time, along with the ability to choose which classifier to use at runtime because it takes into account that one classifier may be more accurate than another in view of the number of samples and their variability;

5. Finally, that ERACI achieved the objectives initially proposed, and the system can be improved through the application of more robust algorithms for the recognition of patterns and through the implementation of a robotic structure capable of enabling autonomous locomotion. These desired characteristics can be implemented in future projects.

REFERENCES

ALVARES, C. A. et al. Köppen's climate classification map for Brazil. **Meteorologische Zeitschrift, Schweizerbart Science Publishers, Stuttgart, Germany**, v. 22, n. 6, p. 711–728, 2013. Retrieved From: <http://dx.doi.org/10.1127/0941-2948/2013/0507>.

BERNARDES, M. S.; BELARDO, G. D. C. Espaçamentos de plantio: Espaçamentos para a cultura de cana-de-açúcar: Manejo nutricional da cultura da cana-de-açúcar. In: **Processos Agrícolas e Mecanização da Cana-de-Açúcar.** [S.l.: s.n.] 2015, 1. Ed. Cassia, Belardo Marcelo Tufaile, R. P. d. S. Guilherme De C. p. 243-258.

CARDOSO, J. R. F. Desenvolvimento de estrutura robótica para aquisição e classificação de imagens (ERACI) de lavoura de cana-de-açúcar. Dissertação (Mestrado) - **Universidade Estadual de Paulista, Faculdade de Ciências Agrárias e Veterinárias, Jaboticabal, SP, Brasil**. 2020. Retrieved From: < http://hdl.handle.net/11449/192961>.

CASTRO, S. G. Q. d. Manejo da adubação nitrogenada em cana-de-açúcar e diagnose por meio de sensores de dossel. Tese (Doutorado) - **Universidade Estadual de Campinas, Faculdade de Engenharia Agrícola**, Campinas, SP, Campinas, Brasil 2016. Retrieved From: <http://repositorio.unicamp.br/jspui/handle/REPOSIP/319248>.

CLIMATE-DATA. **Barretos clima (Brasil)**. Accessed: 2020 October 04. Retrieved From: <https://pt.climate-data.org/america-do-sul/brasil/sao-paulo/barretos-4232/>.

CONAB. **Companhia Nacional de Abastecimento.** Acompanhamento da Safra brasileira de cana-de-açúcar: Safra 2019/20. Accessed: 2020 October 04. Retrieved from: <http://www.conab.gov.br>.

GIMENEZ, L. M.; MOLIN, J. P. Agricultura de Precisão Sob a Perspectiva de Seus Diversos Atores. **Informações Agronômicas, International Plant Nutrition Institute** - IPNI Brasil, 2018.

GONZALEZ, R. C.; WOODS, R. E. Processamento digital de imagens. 3. In: Brasil: **Pearson Education do Brasil**. 2010. ed. São Paulo, S. P.

HAO, Z. et al. Research on driver fatigue detection method based on parallel convolution neural network 2019. In: **IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)**. [S.l.: s.n.], 2019. p. 164-168.

HU, R.;, JIA, W.; LING, H.; HUANG, D. Multiscale distance matrix for fast plant leaf recognition. **IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society**, v. 21, n. 11, p. 4667-4672, 2012.

INPE. **AMBDATA 2020**. Variáveis Ambientais para modelagem de distribuição de Espécies. Mapa de solos. Accessed: 2020 October 04. Retrieved from: <http://www.dpi.inpe.br/Ambdata/mapa_solos.php>.

LANDGE, I. A.; SATOPAY, H. "Secured IoT through hashing using MD5," **2018 Fourth International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)**, Chennai, 2018. p. 1-5, doi: 10.1109/AEEICB.2018.8481007.

LISKI, E. et al. Modeling the productivity of mechanized CTL harvesting with statistical machine learning methods. **International Journal of Forest Engineering**, 1-10, doi: 10.1080/14942119.2020.1820750, 2020, 2020.

LUGER, G. Inteligência artificial. 6. In: **Brasil: Pearson Education do Brasil**. 2013. ed. São Paulo, S. P.

KANG, E.; OH, I. Weak constraint leaf image recognition based on convolutional neural network 2018. In: **International Conference on Electronics, Information, and Communication (ICEIC)**. [S.l.: Sn], 2018. p. 1-4.

KHMAG, A.; AL-HADDAD, S. A. R.; KAMARUDIN, N. Recognition system for leaf images based on its leaf contour and centroid. In: **IEEE 2017 15th Student Conference on Research and Development (SCOReD)**. [S.l.: s.n.], 2017. p. 467-472.

MISHRA, P. K.; DHAR, S.; KALRA, M. K. Landslide detection system using computer vision approach and raspberry pi 2019. In: **International Conference on Communication and Electronics Systems (ICCES)**. [S.l.: s.n.], 2019. p. 1201-1206.

OSROOSH, Y.; KHOT, L. R.; PETERS, R. T. Economical thermal-rgb imaging system for monitoring agricultural crops. **Computers and Electronics in Agriculture**, v. 147, p. 34-43, 2018. ISSN 0168-1699. Retrieved From: <http://www.sciencedirect.com/science/article/pii/S0168169917310360>.

RESENDE, A. V. D.; COELHO, A. M. Amostragem para mapeamento e manejo da fertilidade do solo na abordagem de agricultura de precisão. **Informações Agronômicas, Piracicaba**, v. 1, n. 159, p. 1-8, set. 2017.

RUSTOGI, R.; PRASAD, A. Swift imbalance data classification using smote and extreme learning machine 2019. In: **International Conference on Computational Intelligence in Data Science (ICCIDS)**. [S.l.: s.n.], 2019. p. 1-6.

SAHU, B. K. et al Development of hardware setup of an autonomous robotic vehicle based on computer vision using raspberry pi. In: **Innovations in Power and Advanced Computing Technologies (i-PACT)**. [S.l.: s.n.], v. 1, p. 1-5, 2019.